



---

# 자료구조 (Data Structure)

---

## Programming Assignment 3

## HW 3

### ■ 문제 1: 50점

사용자는 문자 '(', ')', '{', '}', '[', ']' 총 6가지의 괄호만을 입력할 수 있다.

출력은 string이 유효한지 검사하여 true, false를 출력한다.

stack을 이용하여 프로그램을 작성하라.

# HW 3

## ■ 문제 1:

예제	
입력	출력
()	true
{[]}()	true
{}	false
[()]	true
{()}	false

## HW 3

### ■ 문제 1:

#### ◆ 입력:

- 앞서 설명한 6가지 문자만 들어있는 문자열
- 입력의 길이는 최소 2글자, 최대 20글자로 제한

#### ◆ 출력:

- 결과에 따라 true, false를 출력

# HW 3

## ■ 문제 1:

### ◆ 제약 조건:

- 모든 입출력은 stdin, stdout 사용
- 전역변수 사용 가능
- 괄호가 아닌 입력이 있다면 error 출력 후 프로그램 종료
- stack의 ADT가 구현 되어있어야 함

## HW 3

### ■ 문제 2: 100점

프린터는 보통 인쇄 요청이 온 순서대로 인쇄를 진행합니다.  
중요한 문서가 먼저 인쇄가 되도록 하는 프로그램을 만들려고 합니다.

인쇄 대기목록에서 가장 앞에 있는 문서를 대기목록에서 꺼내 나머지 인쇄 대기목록에서 꺼낸 문서보다 중요도가 높은 문서가 있으면 꺼낸 문서를 대기목록에 가장 마지막에 넣습니다. 꺼낸 문서보다 중요도가 높은 문서가 없으면 그 문서를 인쇄합니다.

숫자가 높을 수록 중요도가 높고, 숫자가 작은 수록 중요도가 낮습니다.

중요도의 숫자가 그 문서의 이름입니다.

이 문제에서 출력은 인쇄가 될 때 마다, “인쇄가 된 파일 목록들 / 순서가 바뀐 대기목록” 순으로 문서들을 출력합니다.

queue를 이용하여 구현합니다.

# HW 3

Queue에 하나씩 넣고 앞에 있는 숫자보다 크면 작은 수 pop

## ■ 문제 2:

1 → 5 → 3 → 9 → 6 → 8 끝까지 했으면 가능한 수가 최대.

예제	
입력	출력
1 5 3 9 8 2	9 / 8 2 1 5 3 9 8 / 2 1 5 3 9 8 5 / 3 2 1 9 8 5 3 / 2 1 9 8 5 3 2 / 1 9 8 5 3 2 1 /

## HW 3

### ■ 문제 2:

#### ◆ 입력:

- 각각의 중요도를 갖는 문서의 배열

#### ◆ 출력:

- '/'로 구분해 인쇄가 완료된 문서는 '/'의 좌측에 출력
- 인쇄가 완료되지 않고 인쇄 순서가 바뀐 대기 목록은 '/'의 우측에 출력



## HW 3

### ■ 문제 2:

#### ◆ 제약 조건:

- 모든 입출력은 stdin, stdout 사용
- 전역변수 사용 가능
- queue의 ADT가 구현 되어있어야 함

## HW 3

알고리즘  
→ 앞자리(큰자리)부터 push

→ 차례대로 push하는데 넣을 수보다 이미 stack에 있는거 작으면 제거

### ■ 문제 3: 150점

사용자는 숫자 num과 삭제할 자리 수 k를 입력한다.  
num에서 자리 수 k 만큼 뺐을 때 가장 작은 숫자를 구하는 프로그램을 작성하세요.

예를 들어 num이 13425고 k가 2일 때,  
num에서 2개의 자리를 뺐을 때 가장 작은 숫자는 125이다.

stack을 이용하여 구현하세요.

# HW 3

1432219

count = 7  
k = ~~0~~ ~~2~~ ~~2~~ 0  
top = ~~0~~ 1

stack: | 4 <sup>2</sup> ~~2~~ <sup>2</sup> ~~2~~ <sup>1</sup> 1 9

## ■ 문제 3:

예제	
입력	출력
1432219 3	1219
10200 1	200
10 2	0

## HW 3

### ■ 문제 3:

#### ◆ 입력:

- 첫 번째 줄 : num 입력
- 두 번째 줄 : 입력받은 num에서 뺄 자리의 개수 k

#### ◆ 출력:

- num에서 k만큼의 자리 개수를 뺐을 때 가장 작은 수

# HW 3

## ■ 문제 3:

### ◆ 제약 조건:

- 모든 입출력은 stdin, stdout 사용
- 전역변수 사용 가능
- stack의 ADT가 구현 되어있어야 함

1234

C=123

Count=1

C=12

Count=2

C=1

Count=3

C=0

# 제출 방법

## ■ 소스코드

- ◆ 이름: **HW3\_학번\_문제번호.c**
  - ex) HW3\_20187777\_1.c
- ◆ cpp, txt 파일 등은 **절대** 받지 않음, 반드시 .c 파일로 제출할 것
- ◆ 컴파일 에러가 발생할 경우 0점 처리
- ◆ 무한 루프/세그멘테이션 오류는 해당 testcase 0점 처리
- ◆ 입출력 양식이 틀릴 경우 감점

# 제출 방법

## ■ 보고서

- ◆ 이름: **Document.pdf**
- ◆ 반드시 pdf 파일로 변환 후 제출
- ◆ 대면 수업의 경우 보고서의 hard copy를 제출
- ◆ 그러나, 비대면 수업의 특성상 보고서의 제출을 보류함
- ◆ **hard copy 제출을 하지 않으니 제출 양식이 틀리다면, 무조건 0점 처리**
- ◆ **추후 대면 수업으로 전환 시 제출 방법 변경 사항 공지 예정**

# 제출 방법

## ■ 압축 파일

◆ 이름: **HW3\_학번.zip**

● ex) HW3\_20187777.zip

◆ 압축을 풀면 아래 파일들이 있어야 함.

- HW3\_학번\_1.c
- HW3\_학번\_2.c
- HW3\_학번\_3.pdf
- Document.pdf

◆ 제출 형식이 틀릴 경우 과제 점수의 50% 감점



## 제출 방법

- 11/01 저녁 12:00까지 [sogang.jslee@gmail.com](mailto:sogang.jslee@gmail.com) 으로 제출  
압축파일(HW3\_학번.zip) 제출
- 과제 채점은 gcc compiler 기준
- 명시된 제출 마감 시간보다 늦는 것은 절대 받지 않음
- copy 검사